UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/044,487	01/11/2002	Bodo K. Parady	5181-37601	9276

7590 11/26/2004

Rory D. Rankin
Conly, Rose, & Tayon, P.C.
P.O. Box 398
Austin, TX 78767

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 11/26/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/044,487	PARADY, BODO K.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 November 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *08 November 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-31 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as received on 11/8/2004.

Specification

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Maintained Rejections

4. Applicant has failed to overcome the prior art rejections set forth in the previous Office Action. Consequently, these rejections are respectfully maintained by the examiner and are copied below for applicant's convenience.

Maintained Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-2, 4, 6, and 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Parady, U.S. Patent No. 5,933,627, in view of Kahn, U.S. Patent No. 5,822,788 (herein referred to as Kahn), and further in view of Yoaz et al., "Speculation Techniques for Improving Load Related Instruction Scheduling," May 1999 (herein referred to as Yoaz).

7. Referring to claim 1, Parady has taught a microprocessor comprising:

a) an instruction buffer, wherein said instruction buffer is configured to store a plurality of instructions. See Fig.3, components 102-108.

b) a data cache (Fig.1, component 56, and Fig.2, component 82), wherein said data cache is configured to:

b1) receive a first load address. See Fig.1 and Fig.2 and note that the data cache receives an address from load/store unit 32.

b2) fetch data corresponding to said first load address in response to detecting said data is not present in said data cache. Note from the abstract and column 2, lines 18-24, that an attempt will be made to fetch data from the cache (as is known in the art). However, if the data cache (levels 1 and 2) do not have the data, a miss will occur, and the processor will have to retrieve the data from main memory.

c) Parady has not taught a load prediction coupled to said instruction buffer and to said data cache unit as described in claim 1. However, Kahn has taught a load prediction unit, wherein said load prediction unit is configured to:

c1) detect a first load instruction of said plurality of instructions. Clearly, if a load is being predicted, then it must be detected.

c2) predict a first load address of said first load instruction. See Fig.1 and column 1, lines 26-65, and note that a prefetch address is generated upon predicting a load.

As disclosed in the aforementioned passage, this is useful for hiding memory latency by prefetching the load data to the cache before the load actually executes. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady to include load prediction as taught by Kahn so that memory latency may be hidden.

c3) Parady in view of Kahn has not explicitly taught that the load prediction unit is configured to identify a first instruction of a new thread.

However, Yoaz has taught such a concept. See the section under Fig.3, and note that Yoaz has disclosed that a multithreading architecture, such as the one taught by Parady, would benefit from load prediction in that a thread will be switched if a load instruction is predicted to miss the L2 data cache, thereby not suffering the large latency of accessing main memory. A person of ordinary skill in the art would have recognized that the system of Parady in view of Kahn, with the added capability of thread switching based on a load prediction, would be improved by employing load prediction because if a load-miss were predicted correctly, a new thread would have been fetched and executed earlier than if thread switching based on load prediction were not employed and the system only switched when the actual miss occurs (see the abstract of Parady). Consequently, efficiency would be improved. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady in view of Kahn to include the capability of thread switching based on a load prediction, as taught by Yoaz.

8. Referring to claim 2, Parady in view of Kahn and further in view of Yoaz has taught a microprocessor as described in claim 1.

a) Yoaz has further taught that the load prediction unit comprises circuitry which supports load address prediction and new thread creation. Since Yoaz has taught load prediction, it is inherent that the load predictor includes circuitry which supports load address prediction. In addition, as disclosed, when a load is predicted to miss, a new thread is created (i.e., a new thread is switched in for execution).

b) Furthermore, Kahn has taught a load prediction table with a plurality of entries. See Fig.1.

9. Referring to claim 4, Parady in view of Kahn and further in view of Yoaz has taught a microprocessor as described in claim 2. Kahn has further taught that said plurality of entries in said load prediction table comprises an instruction address field (note the tag field, which corresponds to a specific value of the program counter/instruction address) and an effective address field (see the Prev_Addr field). Kahn has not explicitly taught a valid field. However, Official Notice is taken that valid bits (fields) are well known and expected in the art. More specifically, valid bits indicate whether a particular entry is valid, and consequently, whether the entry may be replaced by another entry (in the case of it being invalid). As a result, in order for the system to determine which entries may be replace, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Kahn's table to include a valid field for each load.

10. Referring to claim 6, Parady in view of Kahn and further in view of Yoaz has taught a microprocessor as described in claim 4. Furthermore, Kahn has taught that said load prediction unit is configured to create a first entry in said load prediction table for said first load instruction

by storing an instruction address in an instruction address field of an entry and storing an effective address in an effective address field of said entry and storing a value in a valid field of said entry which indicates said entry is valid. Clearly, Kahn has taught that if a prediction table exists, then it is populated with information regarding different load instructions. This includes the instruction address (tag) and effective address (prev_addr) fields. Also, note that if a valid field existed in Kahn, which was shown to be an obvious improvement above, then it would have been obvious to mark a newly created entry as valid so that the prediction may be used in the future.

11. Referring to claim 8, Parady in view of Kahn and further in view of Yoaz has taught a microprocessor as described in claim 4. Furthermore, Kahn has taught that said load prediction unit is configured to detect said first entry in said load prediction table by comparing an instruction address of said first detected load instruction to addresses in instruction address fields of said plurality of entries, wherein said instruction address of said first detected load instruction corresponds to said instruction address stored in said instruction address field of said first entry. See Fig. 1 and note that the current value of the program counter is compared to the tags of the table entries. The tags represent the load instruction addresses. In addition, Official Notice is taken that checking a valid field for validity is well known and expected in the art. This is advantageous because the system should not use invalid data for making a prediction. As a result, it would have been obvious to check a valid field for validity to make sure that the prediction is valid.

12. Referring to claim 9, Parady in view of Kahn and further in view of Yoaz has taught a microprocessor as described in claim 6. Furthermore, Kahn has taught that said load prediction

unit is configured to predict a load address of said first detected load instruction by adding a first contents of an effective address field of said first entry to a second contents of a stride field of said first entry. See Fig.1 of Kahn.

13. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over Parady in view of Kahn in view of Yoaz, as applied above, and further in view of Eickemeyer et al., U.S. Patent No. 5,377,336 (herein referred to as Eickemeyer).

14. Referring to claim 3, Parady in view of Kahn and further in view of Yoaz has taught a microprocessor as described in claim 2. Parady in view of Kahn and further in view of Yoaz has not explicitly taught that said circuitry is configured to detect said first load instruction by scanning said plurality of instructions in said instruction buffer for instructions with opcodes which correspond to load instructions. However, Eickemeyer has taught such a concept. See Fig.1, components 105 and 107, and column 3, lines 30-47. Note that the loads are detected early so that prefetching may begin earlier, thereby increasing the chances that the data is available when the load is actually executed. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady in view of Kahn and further in view of Yoaz such that load instructions are scanned based on opcodes. It should be noted that opcodes are what differentiate instructions and therefore, it is the opcodes that the system would be detecting.

15. Claims 5, 7, and 10-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Parady in view of Kahn and further in view of Yoaz, as applied above, and further in view of McFarling et al., U.S. Patent No. 5,758,142 (herein referred to as McFarling).

16. Referring to claim 5, Parady in view of Kahn and further in view of Yoaz has taught a microprocessor as described in claim 4. Kahn has further taught that said load prediction table further comprises a stride field. See Fig.1. Kahn has not taught that the table includes a threshold field. However, McFarling has taught a hit/miss predictor wherein a 2-bit saturating counter (threshold) exists for each load instruction within the program. See column 8, line 50, to column 9, line 33. This counter is incremented if the load hits the cache and decremented if the load misses the cache. The most significant bit is then used to make the prediction (i.e. if the counter equals 10 or 11, a hit is predicted, otherwise a miss is predicted). Therefore, it can be seen that if the load always hits or always misses, the counter will saturate at a binary value of 11 or 00, respectively, thereby allowing the predictor to more accurately predict the outcome of the load in the future. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Kahn to include a threshold field which dictates how a load will be predicted based on past occurrences.

17. Referring to claim 7, Parady in view of Kahn and further in view of Yoaz has taught a microprocessor as described in claim 6. Kahn has further taught that a stride value is stored in a stride field of said entry. See Fig.1. Kahn has not explicitly taught a threshold field of said entry is initialized to indicate no misses for said first load instruction have been recorded. However, McFarling has taught a predictor with such a field. See column 9, lines 23-25. It should be noted that the field tracks the number of hits and misses in the data cache for each instruction.

Clearly, before each instruction accesses the cache for the first time, there will be no record of a hit or miss, and therefore, the field is initialized to indicate no misses. Furthermore, McFarling has taught, in column 9, lines 23-33, that such a field is used to help make a prediction. That is, a past history of the load is referenced to try and determine how it will act in the future. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady in view of Kahn and further in view of Yoaz such that a threshold field exists.

18. Referring to claim 10, Parady in view of Kahn and further in view of Yoaz and further in view of McFarling has taught a microprocessor as described in claim 7. Furthermore, Kahn has taught that said load prediction unit is configured to update an entry of said first detected load instruction of said load prediction table by storing a difference between a received effective address and said contents of said effective address field in said stride field and by storing said received effective address in said effective address field, in response to detecting said detected load instruction hit in said data cache. See Fig.1 of Kahn.

19. Referring to claim 11, Parady in view of Kahn and further in view of Yoaz and further in view of McFarling has taught a microprocessor as described in claim 7. Kahn has further taught said load prediction unit is configured to update an entry of said first detected load instruction of said load prediction table by storing a difference between a received effective address and said contents of said effective address field in said stride field and by storing said received effective address in said effective address field. See Fig.1 of Kahn. Kahn has not taught incrementing contents of a threshold field of said first entry, in response to detecting said detected load instruction missed in said data cache. However, McFarling has taught decrementing the contents of a threshold field, as is known in the art, in response to detecting a load miss. See column 9,

lines 10-22. A person of ordinary skill in the art would have recognized that it is the designer's choice to either increment or decrement a counter when a load miss occurs (applicant increments, McFarling decrements). Whether incrementing or decrementing is chosen, the threshold field would still be used in a similar fashion. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady in view of Kahn and further in view of Yoaz such that a counter is incremented when a load miss occurs.

20. Referring to claim 12, Parady in view of Kahn and further in view of Yoaz and further in view of McFarling has taught a microprocessor as described in claim 5. Yoaz has further taught that said load prediction unit is further configured to predict a load instruction will miss. See the 4th paragraph under Fig.3 of Yoaz.

21. Referring to claim 13, Parady in view of Kahn and further in view of Yoaz and further in view of McFarling has taught a microprocessor as described in claim 5. McFarling has further taught that said load prediction unit is configured to predict said first detected load instruction will miss, in response to detecting contents of a threshold field of said first entry equals a threshold value. See column 9, lines 10-22.

22. Referring to claim 14, Parady in view of Kahn and further in view of Yoaz and further in view of McFarling has taught a microprocessor as described in claim 13. Yoaz has further taught that in response to predicting said first detected load instruction will miss, said load prediction unit is configured to identify a first instruction of a new thread. See the 4th paragraph under Fig.3 of Yoaz.

23. Claims 17-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Parady in view of Kahn and further in view of Yoaz in view of McFarling, as applied above, and further in view of Akkary et al., U.S. Patent No. 6,182,210 (herein referred to as Akkary).

24. Referring to claim 17, Parady in view of Kahn and further in view of Yoaz and further in view of McFarling has taught a microprocessor as described in claim 14. Parady in view of Kahn and further in view of Yoaz and further in view of McFarling has not explicitly taught that said first instruction is said identified by detecting said first instruction is an unconditional branch. However, Akkary has taught that it is good to separate threads at branches, which includes unconditional branches, loops, function calls, etc. See column 5, lines 35-45. Because Akkary has disclosed that a branch is a good place to separate a thread, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady in view of Kahn and further in view of Yoaz and further in view of McFarling such that the first instruction of a new thread is an unconditional branch.

25. Referring to claim 18, Parady in view of Kahn and further in view of Yoaz and further in view of McFarling has taught a microprocessor as described in claim 14. Parady in view of Kahn and further in view of Yoaz and further in view of McFarling has not explicitly taught that said first instruction is said identified by detecting said first instruction immediately follows a loop iteration branch. However, Akkary has taught that it is good to separate threads at loops, branches, function calls, etc. See column 5, lines 35-45. Because Akkary has disclosed that a loop is a good place to separate a thread, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady in view of Kahn and further in view of Yoaz

and further in view of McFarling such that the first instruction of a new thread immediately follows a loop iteration branch.

26. Claims 19 and 21-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Parady, as applied above, in view of Kahn, as applied above, in view of Yoaz, as applied above, and further in view of McFarling, as applied above.

27. Referring to claim 19, Parady has taught a method of thread instruction identification, said method comprising:

a) detecting a first instruction of a plurality of instructions in an instruction buffer is a load instruction. See Fig.3, components 102-108 and the abstract. It is inherent that if a load instruction will be detected when it is in an instruction buffer. That is, before a load operation is performed, a load instruction must be encountered.

b) Parady has not taught predicting a load address for said first instruction, in response to detecting a valid entry exists in a load prediction table for said first instruction. However, Kahn has taught such a concept. See Fig.1 and column 1, lines 26-65, and note that when a load is predicted, a prefetch address is generated (note that if an entry is providing a prediction, then that entry is valid). As disclosed in the aforementioned passage, this is useful for hiding memory latency by prefetching the load data to the cache before the load actually executes. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady to include load prediction so that memory latency may be hidden.

c) Kahn has further taught updating said entry in said load prediction table. See Fig.1 and note that at least the effective address is updated.

d) Parady in view of Kahn has not explicitly taught predicting said first instruction will miss, and in response to said predicting said first instruction will miss, identifying a first instruction of a new thread. However, Yoaz has taught such a concept. See the 4th paragraph under Fig.3. Note that when a load miss is predicted to occur, a thread switch will occur, thereby allowing instructions independent of the predicted load (another thread) to execute instead. This also avoids the large latency of accessing main memory. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady in view of Kahn to identify a new thread in response to a predicted load miss.

e) Parady in view of Kahn and further in view of Yoaz has not explicitly taught that said first instruction is predicted to miss, in response to detecting a miss threshold condition has been met. However, a person of ordinary skill in the art would have recognized that if Yoaz is predicting a load to miss, then some condition must be detected which requires a miss to be performed. McFarling has taught such a condition used with load prediction is the value of a counter. See column 9, lines 20-22, and note that when the counter has a value having an MSB equal to 0, the load is predicted to miss. McFarling has taught that load prediction with a counter is useful because it allows for the scheduling of other instructions which are independent of the missing load instruction. See column 3, lines 7-18. As a result, because McFarling has taught that a counter may be used to predict loads, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady in view of Kahn and further in view of Yoaz such that a load miss is predicted in response to detecting a miss threshold condition has been met.

28. Referring to claim 21, Parady in view of Kahn in view of Yoaz and further in view of McFarling has taught a method as described in claim 19. Furthermore, Kahn has taught that said load prediction unit is configured to detect said first entry in said load prediction table by comparing an instruction address of said first detected load instruction to addresses in instruction address fields of said plurality of entries, wherein said instruction address of said first detected load instruction corresponds to said instruction address stored in said instruction address field of said first entry. See Fig.1 and note that the current value of the program counter is compared to the tags of the table entries. The tags represent the load instruction addresses. In addition, Official Notice is taken that checking a valid field for validity is well known and expected in the art. This is advantageous because the system should not use invalid data for making a prediction. As a result, it would have been obvious to check a valid field for validity to make sure that the prediction is valid.

29. Referring to claim 22 Parady in view of Kahn in view of Yoaz and further in view of McFarling has taught a method as described in claim 19. Furthermore, Kahn has taught that said load prediction unit is configured to create a first entry in said load prediction table for said first load instruction by storing an instruction address in an instruction address field of an entry and storing an effective address in an effective address field of said entry and storing a value in a valid field of said entry which indicates said entry is valid. Clearly, Kahn has taught that if a prediction table exists, then it is populated with information regarding different load instructions. This includes the instruction address (tag) and effective address (prev_addr) fields. Also, note that if a valid field existed in Kahn, which was shown to be an obvious improvement above, then

it would have been obvious to mark a newly created entry as valid so that the prediction may be used in the future.

30. Referring to claim 23, Parady in view of Kahn in view of Yoaz and further in view of McFarling has taught a method as described in claim 22.

a) Kahn has further taught that a stride value is stored in a stride field of said entry. See Fig.1.

b) McFarling has taught initializing a threshold field of said entry to indicate no misses for said first instruction have been recorded. See column 9, lines 23-25. It should be noted that McFarling's field tracks the number of hits and misses in the data cache for each instruction. Clearly, before each instruction accesses the cache for the first time, there will be no record of a hit or miss, and therefore, the field is initialized to indicate no misses. Furthermore, McFarling has taught, in column 9, lines 23-33, that such a field is used to help make a prediction. That is, a past history of the load is referenced to try and determine how it will act in the future. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to include such a threshold field.

31. Referring to claim 24, Parady in view of Kahn in view of Yoaz and further in view of McFarling has taught a method as described in claim 19. Kahn has further taught computing a predicted address for said first instruction, in response to detecting said valid entry does exists, wherein said computing comprises adding a first contents of an effective address field of said valid entry to a second contents of a stride field of said valid entry. See Fig.1.

32. Referring to claim 25, Parady in view of Kahn in view of Yoaz and further in view of McFarling has taught a method as described in claim 19.

a) Kahn has further taught storing a difference between a received effective address and contents of an effective address field of said entry, wherein said difference is stored in a stride field of said entry. See Fig.1

b) Kahn has further taught storing said received effective address in said effective address field of said entry. See Fig.1.

c) McFarling has further taught decrementing (instead of incrementing) contents of a threshold field of said first entry, in response to detecting a valid entry exists in said load prediction table for said first instruction and said first instruction missed. See column 8, line 50, to column 9, line 33. This counter is incremented if the load hits the cache and decremented if the load misses the cache. The most significant bit is then used to make the prediction (i.e. if the counter equals 10 or 11, a hit is predicted, otherwise a miss is predicted). Therefore, it can be seen that if the load always hits or always misses, the counter will saturate at a binary value of 11 or 00, respectively, thereby allowing the predictor to more accurately predict the outcome of the load in the future. A person of ordinary skill in the art would have recognized that it is the designer's choice to either increment or decrement a counter when a load miss occurs (applicant increments, McFarling decrements). Whether incrementing or decrementing is chosen, the threshold field would still be used in a similar fashion. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady in view of Kahn in view of Yoaz and further in view of McFarling such that a counter is incremented (as opposed to decremented) when a load miss occurs.

33. Referring to claim 26, Parady in view of Kahn in view of Yoaz and further in view of McFarling has taught a method as described in claim 19. Kahn has further taught said updating comprises:

a) storing a difference between a received effective address and contents of an effective address field of said entry, wherein said difference is stored in a stride field of said entry. See Fig.1.

b) storing a received effective address in said effective address field, in response to detecting a valid entry exists in said load prediction table for said first instruction and said first instruction hit. See Fig.1. Kahn has not disclosed that the effective address field is not updated in all situations (hit of miss), whereas the table in Fig 1 shows that the effective address field is updated with each access since the field stores the previous address. Therefore, the table is updated on a load hit.

34. Referring to claim 27, Parady in view of Kahn in view of Yoaz and further in view of McFarling has taught a method as described in claim 19. McFarling has further taught said predicting said first load instruction will miss comprises detecting a value in a threshold field of said entry equals a threshold value. See column 9, lines 10-22. Note that if the MSB of an M-bit counter is 0, then a load is predicted to miss. Therefore, if a counter is two bits wide, the load will be predicted to miss when the counter has a value of 01 or lower, whereas if a counter is three bits wide, then the load will be predicted to miss when the counter has a value of 011 or lower.

35. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Parady in view of Kahn in view of Yoaz in view of McFarling, as applied above, and further in view of Eickemeyer, as applied above.

36. Referring to claim 20, Parady in view of Kahn in view of Yoaz and further in view of McFarling has taught a method as described in claim 19. Parady in view of Yoaz and further in view of Kahn has not explicitly taught that said circuitry is configured to detect said first load instruction by scanning said plurality of instructions in said instruction buffer for instructions with opcodes which correspond to load instructions. However, Eickemeyer has taught such a concept. See Fig.1, components 105 and 107, and column 3, lines 30-47. Note that the loads are detected early so that prefetching may begin earlier, thereby increasing the chances that the data is available when the load is actually executed. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady in view of Kahn in view of Yoaz and further in view of McFarling such that load instructions are scanned based on opcodes. It should be noted that opcodes are what differentiate instructions and therefore, it is the opcodes that the system would be detecting.

37. Claims 30-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Parady in view of Kahn in view of Yoaz in view of McFarling, as applied above, and further in view of Akkary, as applied above.

38. Referring to claim 30, Parady in view of Kahn in view of Yoaz and further in view of McFarling has taught a method as described in claim 19. Parady in view of Kahn in view of Yoaz and further in view of McFarling has not explicitly taught the concept of claim 30.

However, Akkary has taught that said identifying said first thread instruction comprises detecting a second instruction of a second plurality of instructions in said instruction buffer is an unconditional branch, and selecting said second instruction as said first thread instruction, wherein said second plurality of instructions in said instruction buffer are subsequent in program order to said first instruction. See column 5, lines 35-53, and note that Akkary has taught that it is good to separate threads at branches, which includes unconditional branches, loops, function calls, etc., and that a program is broken up in order. Because Akkary has disclosed that a branch is a good place to separate a thread, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady in view of Kahn in view of Yoaz and further in view of McFarling such that the first instruction of a new thread is an unconditional branch.

39. Referring to claim 31, Parady in view of Kahn in view of Yoaz and further in view of McFarling has taught a method as described in claim 19. Parady in view of Kahn in view of Yoaz and further in view of McFarling has not explicitly taught the concept of claim 31. However, Akkary has taught that said identifying said first thread instruction comprises detecting a second instruction of a second plurality of instructions in said instruction buffer immediately follows a loop iteration branch, and selecting said second instruction as said first thread instruction, wherein said second plurality of instructions in said instruction buffer are subsequent in program order to said first instruction. See column 5, lines 35-53, and note that Akkary has taught that it is good to separate threads at loops, branches, function calls, etc., and that the program is broken up in order. See column 5, lines 35-45. Because Akkary has disclosed that a loop is a good place to separate a thread, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Parady in view of Kahn in view of Yoaz and further

in view of McFarling such that the first instruction of a new thread immediately follows a loop iteration branch.

Allowable Subject Matter

40. Claims 15-16 and 28-29 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Response to Arguments

41. Applicant has argued, using 35 U.S.C. 103(c), that Parady, U.S. Patent No. 5,933,627, is invalid prior art because it is a same-assignee 102(e) reference used in a 103 rejection. However, the examiner asserts that Parady is in fact valid because it qualifies as a 102(b) reference and not a 102(e) reference, as applicant suggests. More specifically, applicant's application was filed on January 11, 2002. It also claims priority to a provisional application filed on January 11, 2001, which in turn does not claim priority to something else. Therefore, the effective filing date of applicant's application is January 11, 2001. However, the Parady reference was published on August 3, 1999, well over a year before applicant's filing date. Consequently, Parady qualifies as a 102(b) reference, thereby making it valid prior art when used in a 103 rejection.

Conclusion

42. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH

David J. Huisman
November 19, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100